IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 10/693,409
Filing Date ..... October 24, 2003
Confirmation No. 1913
Inventorship ..... Snover et al.
Appellant ..... Microsoft Corp.
Group Art Unit 2193
Examiner ..... Wood, William H.
Attorney's Docket No. MS1-1739US
Title: Mechanism for Providing Extended Functionality to Command Line Instructions

APPEAL BRIEF

To: Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: Robert G. Hartman (Tel. 509-324-9256 x265; Fax 509-323-8979)
Customer No. 22801

Pursuant to 37 C.F.R. §41.37, Appellant hereby submits an appeal brief for application 10/693,409, filed October 24, 2003, accompanied with a one-month extension of time. Accordingly, Appellant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

(1) Real Party in Interest

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject Application.

(2) Related Appeals and Interferences

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

(3) Status of Claims

Claims 1-20 stand rejected, are pending in the Application, and are set forth in the Appendix of Appealed Claims on page 20.

(4) Status of Amendments

A first Office Action was issued on February 7, 2006.

A Response was filed on May 5, 2006. Claims 1, 8, 9, and 16 were amended.

A Final Office Action was issued on July 24, 2006.

A Response was filed on September 29, 2006. No claims were amended.

An Advisory Action was issued on November 2, 2006.

A First Notice of Appeal was filed on November 24, 2006.

A Non-Final Office Action was issued on June 5, 2007.

A Second Notice of Appeal, accompanied by a Request to Reinstate an Appeal, was filed on June 27, 2007.

(5) Summary of Claimed Subject Matter

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters, if any. These specific reference characters are examples of particular elements of the drawings for certain embodiments of the claimed subject matter and the claims are not limited to solely the elements corresponding to these reference characters.

With regards to claim 1, in a command line operating environment, a computer-executable method comprises executing each command on a command line in a first execution mode (Fig. 17, Block 1706; page 55, line 19 through page 56, line 4) or in an alternate execution mode (Fig. 17, Blocks 1710-1712; page 55, line 19 through page 56, line 4; page 56, line 17 through page 60, line 16), wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode (Fig. 17, "Yes" branch of Block 1704; page 55, line 19 through page 56, line 4; page 56, lines 17-18), the alternate execution mode being provided by the command line operating environment (page 56, lines 1-3).

With regard to claim 9, at least one computer-readable medium has computer-executable instructions for performing a method, comprising receiving a command line operative by a command line operating environment that directs the performance of a task; determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode (page 2, lines 1-4; page 4, 18-24; page 55, line 19 through page 56, line 3; page 57, line 18 through page 58, line 6; page 80, lines 7-9); if the parameter is present, simulating the performance of the task by the command line operating

environment (page 2, lines 1-4; page 4, 18-24; page 55, line 19 through page 56, line 3; page 57, line 18 through page 58, line 6; page 80, lines 7-9); and reporting the results of the simulation by the command line operating environment (page 2, lines 1-4; page 4, 18-24; page 55, line 19 through page 56, line 3; page 57, line 18 through page 58, line 6; page 80, lines 7-9).

With regard to claim 16, a system provides a command line operating environment, the system comprising a processor (Fig. 1, element 102; page 1, lines 15-18); and a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor (Fig. 1, element 104; page 1, lines 15-18), the computer-executable instructions performing a method comprising executing each command entered on a command line, wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality (page 4, lines 18-25; page 80, lines 2-12).

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1-3, 6, 8-10, and 16-18 stand rejected under 35 U.S.C. §102(b) as being anticipated by Altiris RapidInstall, version 3.0, "Release Notes" (hereinafter, "RapidInstall").

Claims 7 and 11 stand rejected under 35 U.S.C. §102(b) as anticipated by or, in the alternative, under 35 U.S.C. §103(a) as being obvious over RapidInstall.

Claims 4 and 19 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of R. Dykhuis, "Beefing up DOS with 4DOS" (hereinafter, "Dykhuis").

Claims 5 and 20 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of T. Yager, "Taking Command of Windows NT" (hereinafter, "Yager").

Claims 13-15 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of U.S. Patent No. 7,103,590 to Murthy et al. (hereinafter, "Murthy").

(7) Argument

A. The rejections under 35 U.S.C. §102(b) fail to establish that RapidInstall anticipates the claims against which it is cited.

Appellant respectfully submits that the Office has not established that the RapidInstall reference anticipates the claims rejected under 35 U.S.C. §102(b). The discussion begins with a section entitled "The §102 Standard", which describes the standard by which claim anticipation is established. Next, a section entitled "Appellant's Disclosure" describes salient aspects of the present Application. A section entitled "The RapidInstall Reference" follows and explains the RapidInstall reference. Finally, a section entitled "The Claims" presents Appellant's reasoning as to why the Office has not established that RapidInstall anticipates the rejected claims.

The §102 Standard

Appellant first notes the requirements of MPEP §2131, which states that to anticipate a claim, the reference must teach "every element" of the claim. This section further states that:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).... "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an *ipsissimis verbis* test, i.e., identity of terminology is not required. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Accordingly, the reference must contain, within its four corners, exactly the subject matter of the claim, as it appears in the claim, in order to support a valid finding of anticipation. The absence from a cited §102 reference of ***any*** claimed element negates a finding of anticipation. *See, e.g., Kloster Speedsteel AB, et al. v. Crucible, Inc.*, et al., 793 F.2d 1565, 230 USPQ 81 (Fed. Cir. 1986) (emphasis added).

Appellant's Disclosure

Having discussed the standard against which the Office's rejections must be measured, Appellant now provides a brief description of the current Application. Appellant notes that while this description is provided to assist the reader in appreciating the subject matter described in Appellant's specification, it is not intended to imply or impose specific limitations with regard to Appellant's claims.

Perhaps a good place to start to gain an appreciation of the claimed subject matter is in Appellant's "Background" section. There, the "Background" section describes that in a command line environment, a command line interface allows a user to directly perform a task by entering in a command. For example, a command line interface may be invoked that provides a window that displays a prompt (e.g., "C:\>"). A user may type in any of numerous commands, such as "dir", at the prompt to perform the command. *Appellant's Specification*, Background.

With this setting in mind, the present Application provides extended functionality to command line instructions in a manner that is significantly different from the traditional mechanisms for extending functionality. For example, in traditional mechanisms, each command that desires the extended functionality must ***incorporate the corresponding code into the command itself.*** *See, e.g., RapidInstall infra.* The command itself must then parse the command string to determine whether a switch (e.g., verbose, whatif) was provided and execute the extended functionality accordingly. ***In contrast***, mechanisms described within the present Application allow users to specify an argument within the command string in order to execute the extended functionality for a particular cmdlet, as long as the cmdlet incorporates a hook to the extended functionality. That is, an operating environment itself provides the extended functionality—and not merely the command itself. Thus, the present mechanisms *minimize the amount of code system administrators need to write.* In addition, the *extended functionality is implemented in a uniform manner. Appellant's Specification*, p. 74, lines 7-20. (emphasis added).

The RapidInstall Reference

RapidInstall, meanwhile, describes a traditional mechanism as discussed above. Functionally, the RapidInstall application captures changes that occur on a personal computer when the user installs an application or makes configuration changes. RapidInstall then captures and builds these changes into a compressed, self-contained, and self-extracting ***executable (.exe) file called a RapidInstall Package (or "RIP").*** This RIP file can then be loaded onto client personal computers to make for easier installation of the new applications and configuration changes.

RapidInstall also discusses various command line switches (e.g., "-r", "-cu", "-si") that can be executed from a command line. However, importantly, the RapidInstall program itself incorporates the code that handles these switches and functionality. One specific command line switch provided by RapidInstall is –si, which allows for a "Simulated Install". Here, a user may simulate the installation of a built RIP file to determine if the user may experience any potential problems upon actual installation of the RIP file. Again, the user may use the "–si command line" to view this simulation. When the user runs the simulation via the –si command line, the RIP runs without actually making changes to the system, and a log flags any possible deployment problems. *RapidInstall 3.0 Release Notes*, p. 1. Again, the –si command *itself* provides this simulation functionality.

The Claims

Claim 1

Claim 1 recites in a command line operating environment, a computer-executable method comprising (emphasis added):

- executing each command on a command line in a first execution mode or in an alternate execution mode, wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, ***the alternate execution mode being provided by the command line operating environment.***

In making out a rejection of this claim in both the Final Office Action and the recent Non-Final Office Action, the Office merely cites to a single portion of RapidInstall, namely "page 1, section 'Simulated Install –si command line". *Non-Final Office Action mailed 06/05/2007*, p. 2-3. In the "Response to Arguments" section of the Final Office Action mailed July 24, 2006, meanwhile, the Office also states that "RapidInstall is a command line operating environment under the broadest reasonable interpretation (page 1, last paragraph, "-si command line"). Clearly, RapidInstall provide[s] command line functionality and is an environment/program/software, which operates. The claims do not require more. Therefore, as a command line operating environment, RapidInstall provides the above limitations in question." *Final Office Action mailed 07/24/2006*, p. 8.

Appellant respectfully disagrees and submits that RapidInstall fails to anticipate Appellant's claim 1. Specifically, Appellant respectfully submits that RapidInstall at least fails to disclose, teach, or suggest an "alternate execution mode *being provided by the command line operating environment*", as recited in Appellant's claim. (emphasis added).

To understand the importance of this claim element, Appellant first directs the reader's attention to the section entitled "Appellant's Disclosure" above. In brief, this section states that traditional mechanisms for providing extended functionality to command line instructions require that each command that desires the extended functionality ***actually incorporate the code into the command itself***, as opposed to "being provided by the command line operating environment". As shown below, the RapidInstall reference merely incorporates its functionality into the code itself in accordance with traditional mechanisms.

In order to show that RapidInstall fails to disclose, teach, or suggest the language of Appellant's claim, Appellant begins by examining the well-known and well-defined term "operating environment". Appellant respectfully submits that this term has a clear and unmistakable meaning to one skilled in the art. Specifically, the term "operating environment" refers to "***the environment in which users run programs***". (See definition obtained from Webopedia in Evidence Appendix (9), pages 25-26). In some instances, an operating environment comprises a command line operating environment, such as MS-DOS or Unix Shell. Furthermore, an "operating environment" is often deemed a control program, which itself is defined as "a program that enhances an operating system by ***creating an environment in which you can run other programs***". (See definition obtained from Webopedia in Evidence Appendix (9), page 27).

In contrast, an executable file, such as the RIP file discussed above, is merely a program file that is ready to run in a particular environment. As such, an executable file *runs within* an operating environment—but is *not itself* an operating environment.

With this vital distinction in mind, Appellant now turns to the RapidInstall reference. As discussed above, the RapidInstall reference provides for a "Simulated Install" of an RIP file (i.e., ***an executable file***). This simulation allows a user to simulate the installation of a built RIP file in order to determine if the user will experience any potential problems upon actual installation of the RIP file. To do so, the user may use the "–si command line". *RapidInstall 3.0 Release Notes*, p. 1. Again, the Office currently cites to this "-si command line" as disclosing Appellant's "command line operating environment".

Importantly, however, nothing in RapidInstall relates to the providing of any operating environment. Instead, RapidInstall is silent on the matter. Appellant respectfully submits, however, that this silence is quite coherent and understandable, as both the executable RIP file and the simulated command line (-si) execute *within* a traditional command line operating environment. That is, both the RIP file as well as its simulation *runs within and on* a traditional command line operating environment. As such, the –si command *itself* provides for the simulation, in contrast with a simulation "*being provided by the command line operating environment*", as recited in Appellant's claim. (emphasis added). Stated otherwise, RapidInstall actually discloses using a traditional command line environment where *commands (i.e., executable files) incorporate all of the desired functionality*. Appellant's claim, meanwhile, recites that extended functionality

(e.g. an alternate execution mode) is "*provided by the command line operating environment*" *itself*. (emphasis added).

To further highlight this vital distinction, imagine that a RapidInstall user wishes to simulate the installation of an RIP file. This user would first enter the -si command into a command line operating environment. *Based on the command itself*, RapidInstall would then simulate the results of running the RIP file. Importantly, if the user wished to view the simulation of a *second* installation of a file *unrelated* to the RIP file, the traditional command line operating environment would be unable to do so. Furthermore, because the –si command of RapidInstall only relates to simulating the RIP file, RapidInstall would similarly not help the user in viewing simulated results of the second installation. As discussed at length above, this failing of RapidInstall is due to the fact that the simulation functionality is provided by the *–si command itself—and is not "being provided by the command line operating environment*", as recited in Appellant's claim. (emphasis added).

Appellant therefore respectfully submits that the RapidInstall reference fails to support a *prima facie* case of anticipation, as RapidInstall has not and cannot be shown to disclose, or teach or suggest, "wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, *the alternate execution mode being provided by the command line operating environment*", as recited in Appellant's claim. (emphasis added).

For at least this reason, Appellant respectfully submits that this claim stands allowable.

Additionally, Appellant also takes this space to address the Examiner's "Response to Arguments" section of the outstanding Non-Final Office Action. Here, the Examiner states the following in regards to Appellant's arguments above:

> The phrase "provided by the command line operating environment" is broad and includes all functionality provided by that environment. It is not clear the exact relationship of how the functionality is provided by that environment. Thus, RapidInstall being executed "in", "on" or "by" the command line operating environment is functionality provided by the command line operating environment. This is consistent with the originally filed disclosure and Appellant's recently cited Webopedia definition.... Accordingly in DOS for example, the command line operating environment would include[] "hardwired" commands as well as batch commands and executable add-on commands.
>
> *Non-Final Office Action mailed 06/05/2007*, p. 8-9.

Appellant respectfully but quite strongly disagrees with the Examiner's recent interpretation of Appellant's claim. Appellant submits that it would strain the English language to conclude that a traditional command line operating environment *provides* the "alternate execution mode", as recited in Appellant's claim. At most, the traditional command line operating environment, such as the command line operating environment in which the RIP file executes, merely *enables* execution of the RIP file.

Appellant respectfully submits that "enabling" an action substantially and distinctly differs from "providing" the functionality of an action. While enabling is defined as "allowing" an action, "providing" is defined as actually "supplying" the action's functionality. ROGET'S NEW MILLENNIUM THESAURUS, http://thesaurus.reference.com/browse/enable, http://thesaurus.reference.com/

browse/provide, (1st ed. 2007). With this important distinction in mind, Appellant respectfully submits that the Office fails to show how these two disparate terms should be deemed synonymous. Additionally, Appellant respectfully submits that this showing cannot be made, as the traditional command line operating environment of the RapidInstall reference at most enables the RIP file's functionality—and fails to affirmatively "provide" or "supply" this functionality.

In sum, Appellant once more respectfully submits that the RapidInstall reference fails to disclose, teach, or suggest "wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, *the alternate execution mode being provided by the command line operating environment*", as recited in Appellant's claim. (emphasis added).

Claims 2-8

Claims 2-8 depend from claim 1 and, as such, the remarks made above in regards to claim 1 apply equally to these claims. The rejections of these claims are also improper as failing to disclose these claims' own recited features which, in combination with those recited in claim 1, are not shown to be disclosed, taught, or suggested in the reference of record. Appellant further notes that while some of these dependent claims are rejected under §103(a) in view of RapidInstall and a secondary reference, the secondary references are not cited to teach, nor do they teach, the elements missing in the RapidInstall reference. As such, the addition of these references is not seen to add anything of substance to the rejection of the base claim.

Claim 9

Claim 9 recites at least one computer-readable medium having computer-executable instructions for performing a method, comprising (emphasis added):

- receiving a command line operative by a command line operating environment that directs the performance of a task;
- determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode;
- if the parameter is present, ***simulating the performance of the task by the command line operating environment***; and
- ***reporting the results of the simulation by the command line operating environment.***

In making out a rejection of this claim, the Office states that this claim is "substantially the same" as claim 1 and, as such, is rejected "in the same manner" as claim 1. *Non-Final Office Action mailed 06/05/2007*, p. 4. Therefore, Appellant respectfully submits that this claim stands allowable for at least the reasons discussed above in regards to claim 1. For example, Appellant respectfully submits that RapidInstall at least fails to disclose if the parameter is present, *simulating the performance of the task by the command line operating environment*", as recited in Appellant's claim. (emphasis added). Furthermore, Appellant respectfully submits that RapidInstall also fails to disclose "if the parameter is present, *simulating the performance of the task by the command line operating environment*". (emphasis added).

Claim 10-15

Claims 10-15 depend from claim 9 and, as such, the remarks made above in regards to claim 9 apply equally to these claims. The rejections of these claims are also improper as failing to disclose these claims' own recited features which, in combination with those recited in claim 9, are not shown to be disclosed in the reference of record. Appellant further notes that while some of these dependent claims are rejected under §103(a) in view of RapidInstall and a secondary reference, the secondary references are not cited to teach, nor do they teach, the elements missing in the RapidInstall reference. As such, the addition of these references is not seen to add anything of substance to the rejection of the base claim.

Claim 16

Claim 16 recites a system that provides a command line operating environment, the system comprising (emphasis added):

- a processor; and
- a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor, the computer-executable instructions performing a method comprising:
- executing each command entered on a command line, *wherein if the command includes an instruction to execute the command using* ***extended functionality provided by the command line operating environment****, executing the command using the extended functionality.*

In making out a rejection of this claim, the Office states that this claim is "substantially the same" as claim 1 and, as such, is rejected "in the same manner" as claim 1. *Non-Final Office Action mailed 06/05/2007*, p. 4. Therefore, Appellant respectfully submits that this claim stands allowable for at least the reasons discussed above in regards to claim 1. For example, Appellant respectfully submits that RapidInstall at least fails to disclose "executing each command entered on a command line, *wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality*". (emphasis added).

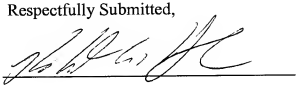
Claims 17-20

Claims 17-20 depend from claim 16 and, as such, the remarks made above in regards to claim 16 apply equally to these claims. The rejections of these claims are also improper as failing to disclose these claims' own recited features which, in combination with those recited in claim 16, are not shown to be disclosed in the reference of record. Appellant further notes that while some of these dependent claims are rejected under §103(a) in view of RapidInstall and a secondary reference, the secondary references are not cited to teach, nor do they teach, the elements missing in the RapidInstall reference. As such, the addition of these references is not seen to add anything of substance to the rejection of the base claim.

Conclusion

The Office has not established a *prima facie* case of anticipation and/or obviousness for at least the reasons discussed above. Accordingly, Appellant respectfully requests that the rejections be overturned and that the pending claims be allowed to issue.

Respectfully Submitted,

Dated: 2007/09/27 By: 

Robert G. Hartman
Lee & Hayes, PLLC
Reg. No. 58,970
(509) 324-9256 ext. 265

(8) Appendix of Appealed Claims

1. (Previously Amended) In a command line operating environment, a computer-executable method comprising:

executing each command on a command line in a first execution mode or in an alternate execution mode, wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, the alternate execution mode being provided by the command line operating environment.

2. (Original) The computer-executable method of claim 1, wherein the alternate execution mode visually displays results of executing the command.

3. (Original) The computer-executable method of claim 1, wherein the alternate execution mode visually displays simulated results of executing the command.

4. (Original) The computer-executable method of claim 1, wherein the alternate execution mode prompts for verification of executing the command before executing the command.

5. (Original) The computer-executable method of claim 1, wherein the alternate execution mode performs a security check to determine whether a user requesting the execution of the command has sufficient privileges to execute the command.

6. (Original) The computer-executable method of claim 1, wherein executing the command in the alternate execution mode further occurs when the command line includes a switch indicating the alternate execution mode.

7. (Original) The computer-executable method of claim 6, wherein the switch comprises "whatif" and the alternate execution mode visually displays simulated results of executing the command.

8. (Previously Amended) The computer-executable method of claim 1, wherein the instruction comprises a call to a method provided by the command line operating environment.

9. (Previously Amended) At least one computer-readable medium having computer-executable instructions for performing a method, comprising:

receiving a command line operative by a command line operating environment that directs the performance of a task;

determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode;

if the parameter is present, simulating the performance of the task by the command line operating environment; and

reporting the results of the simulation by the command line operating environment.

10. (Original) The computer-readable medium of claim 9, wherein the parameter comprises a switch.

11. (Original) The computer-readable medium of claim 10, wherein the switch comprises "whatif".

12. (Original) The computer-readable medium of claim 9, wherein the task comprises a stand-alone executable command.

13. (Original) The computer-readable medium of claim 9, wherein the task comprises a pipeline of executable commands, each executable command operating in a separate process.

14. (Original) The computer-readable medium of claim 9, wherein the task comprises a pipeline of executable commands, each executable command operating in the same process.

15. (Original) The computer-readable medium of claim 14, wherein each executable command comprises an instantiated class.

16. (Previously Amended) A system that provides a command line operating environment, the system comprising:

a processor; and

a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor, the computer-executable instructions performing a method comprising:

executing each command entered on a command line, wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality.

17. (Original) The system of claim 16, wherein the extended functionality comprises visually displaying results of executing the command.

18. (Original) The system of claim 16, wherein the extended functionality comprises visually displaying simulated results of executing the command.

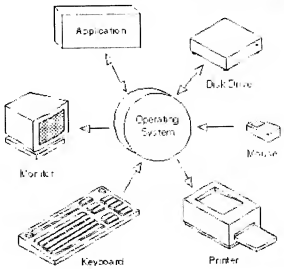
19. (Original) The system of claim 16, wherein the extended functionality comprises prompting for verification before executing the command.

20. (Original) The system of claim 16, wherein the extended functionality comprises performing a security check to determine whether a user requesting the execution of the command has sufficient privileges to execute the command.

(9) **Evidence appendix:** The following evidence was entered into the record by the Office in the Advisory Action dated November 2, 2006.

operating environment

Last modified: Sunday, September 01, 1996



The environment in which users run programs. For example, the DOS environment consists of all the DOS commands available to users. The Macintosh environment, on the other hand, is a graphical user interface that uses icons and menus instead of commands.

There is a thin line between operating environments and shells. Historically, shells are the interfaces to operating systems. They do not actually add any new capabilities; they simply provide a better user interface. So-called intelligent shells, however, actually extend an

operating system's capabilities, so there is little difference between intelligent shells and operating environments.

Operating environments are sometimes called control programs.

For internet.com pages about operating environment CLICK HERE. Also check out the following links!

LINKS = Great Page!

Related Categories

Operating Systems

Related Terms

control program

environment

graphical user interface

JVM

Microsoft Windows

operating system

shell

virtual machine

MENU
Home
Term of the Day
New Terms
Pronunciation
New Links
Quick Reference
Did You Know?
Categories
Tech Support
Webopedia Jobs
About Us
Link to Us
Advertising

Compare Prices: go
HardwareCentral

Talk To Us...
Submit a URL
Suggest a Term
Report an Error

control program

Last modified: Sunday, September 01, 1996

(1) A program that enhances an operating system by creating an environment in which you can run other programs. Control programs generally provide a graphical interface and enable you to run several programs at once in different windows.

Control programs are also called *operating environments*.

(2) Another term for *operating system*.

(10) Related proceedings appendix. None